2023 European Conference on Computing in Construction
40th International CIB W78 Conference
Heraklion, Crete, Greece
July 10-12, 2023

AUTOMATED SUPPLEMENT OF INFORMATION REQUIREMENTS FOR TENDERING DATA

Martina Mellenthin Filardo[1], Leon Heiko Wiesner[2], Jürgen Melzner[1], and Hans-Joachim Bargstädt[1]
[1]Bauhaus-Universität Weimar, Weimar, Germany
[2]Pbr Planungsbüro Rohling AG, Jena, Germany

Abstract

This paper examines data from information exchange requirements (IER) using a two-part case study, in which Germany-specific tendering, assignment and invoicing data (TAI) was attributed to a model for selected elements automatically using IER and self-developed algorithms, where the attributed object is connected to partial services in the quantity take-off (QTO). A comparison of determined quantities from the calculating software with the calculated quantities from the authoring tool validated the automated mapping of TAI data to the modelled objects and the items in the QTO. The result is a user-friendly attribution option for digital twins with TAI-compatible data.

Introduction

Thanks to current data collection and analysis, digital methods in construction allow for a comprehensive array of simulations, surveys and assessments (Borrmann et al., 2021; Abbaspour, 2021). Possible applications of digital methods, such as Building Information Modelling (BIM) (Eastman et al., 2018), are presented in so-called BIM use cases, which represent the BIM services in a specific project phase as well as the operational implementation of the BIM method in a construction project (Deubel et al., 2018). They must be planned in advance for their correct use and outcome and should be limited to processes in which the application of digital methods makes sense and provides increased added value compared to its implementation effort.

According to (Abbaspour, 2021; Schwerdtner, 2018), BIM is frequently applied in projects only in the design and construction phases, often separately, while the benefits in the operational phase are little exploited, even though the method itself is life cycle-oriented. When using the BIM method, its' components with contractual relevance (Roquette and Schweiger, 2020), such as the Employer Information Requirements (EIR) must be addressed (Mellenthin Filardo and Krischler, 2020; Abbaspour, 2021). EIR documents define project deliverables for specific use cases and support the information transfer between life cycle phases. Within the mostly paper-based EIR documents, the concept of Exchange Information Requirements (also abbreviated with EIR) must be addressed. To avoid misunderstandings, the authors delimit both concepts: EIR stands for Employer Information Requirements while the other abbreviation is based on the definition by (VDI Verein Deutscher Ingenieure e.V., 2021b), establishing the IER, which stand for Information Exchange Requirements. The latter are discussed in particular, as they represent the machine-readable, information-centered part of Employer Information Requirements, that deals with data interfaces between models and the general transferability of necessary information from one model to the subsequent model or environment. In parallel, the associated upstream and downstream processes or content components of an EIR, that are directly related to the IER, are elaborated.

Methodology

With the motivation to employ use cases and thus EIRs and its' respective machine-readable component IER as automated specifications for concrete BIM services, a set of workflows is defined to represent the operational implementation of the method and provide a concept for data interfaces to guarantee a smooth implementation of chosen BIM use cases into a project, as shown in Figure 1. Components of the EIR that have no connection with the IER are excluded.

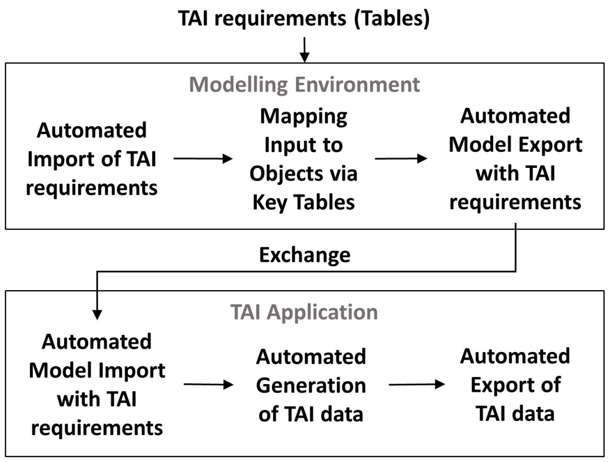

Figure 1: Overview of the proposed workflow.

After a short introduction to relevant concepts and tools, the first implementation is presented, where the IERs for the chosen practical project are formulated for the (German) BIM use case 23: Quantity and Cost Calculation 5D Planning (Deubel et al., 2018). In its implementation, specific elements within a model are automatically attributed according to the (German) Standard Service Specification for Construction (see section Standard Service Specifications in Construction), defined in the generated IERs, so that they can later be unambiguously assigned to a service specification item (service directory item) with a quantity and an associated unit price in the tender using a key parameter. For the case study, drywall walls were modelled with all TAI-related properties, according to the German TAI-classification group 039 ”Drywall Work Walls” (StLB-Bau) formulated in the IERs, automatically through a modular script. With this, further attributions of different elements in the model are possible. The objective of this work is to investigate the interfaces between the BIM use cases and different data exchange requirements resulting from the data models and digital twins of varying project stakeholders. In this context, special emphasis is given to the practical implementation of a specific BIM use case in a real construction project, specifically as to how these can be implemented in applications or how the transferability of the data requirements from one application to a subsequent application can look like. In connection with the data interfaces and the associated data exchange requirements of the models, the individual interface concepts are discussed in said set of workflows, which offer the advantage that they represent a direct implementability of the BIM use cases and are applicable to the project. The presented set of workflows was designed to be modular, allowing for higher re-usability and modularity of the set as well as its individual components within the data exchange process.

The proposed set of workflow(s) was tested with input data from a practical project, a modernisation of a cultural centre near Leipzig, Germany, which was modelled using Autdesk Revit as an authoring tool. The IER information, that needs to be transferred into the model, was available as spreadsheets.

Background

In the following subsections, an introduction to relevant topics as well as the used components and tools is given to aid the understanding of its' implementation (see section Implementation).

Standard Service Specifications in Construction

In Germany, the so-called ”StLB-Bau” (Standardleistungsbuch für das Bauwesen) was established to define service specifications. In free translation, it stands for the Standard Service Book for the Construction Industry, is maintained by the Gemeinsamen Ausschuss Elektronik im Bauwesen (GAEB), which is made up of representatives of public and private clients, architects, engineers and the construction industry in conjunction with the German Contracting Committee for Construction Services (DVA) and published by the German Institute for Standardization (DIN) (GAEB - Gemeinsamer Ausschuss Elektronik im Bauwesen, 2020). It is used for the rational description of construction services and the exchange of information between involved partners in a construction project. By describing individual services in accordance with the StLB-Bau, imprecise texts are avoided, and the statements are correctly understood by construction experts. The StLB-Bau is hierarchically structured in service areas, contains standard descriptions, i.e., specified texts for general provisions for the description of services and for additional technical regulations, as well as standard service descriptions, e.g., specified texts for the description of services or partial services, regarding information on the type of construction, component, building material and dimension for the manufacturing process and the quality of a service. The StLB-Bau text modules can be extended or added, if necessary. The description of (partial) services includes an identification of the area the service is to be performed in (usually a number), a unique ID, a unit (for measuring and/or calculating), a long description text of the service as well as a short description, and a description supplement, if necessary (Rösel and Busch, 2008).

Information Requirements

To ensure that transferred branch-specific and partial BIM models do not stand out of context with the entire project or that models can be used for subsequent life cycle phases, the definition of project-specific EIRs is advisable. EIR documents are regarded as additional contract documents in tendering and provide specifications for the models (services) to be delivered in order to set concrete requirements for the (planning) process. They must contain performance descriptions and have priority over the BIM Execution Plan (BEP). Contents of EIR documents are defined in (British Standards Institution, 2013; ARGE BIM4INFRA2020, 2019) and analysed in (Mellenthin Filardo, 2019). Interfaces for data import and export must be implemented without errors, which places great demand on vendors. Moreover, software applications often require additional manual configuration for data transfer. Therefore, it is important to precisely describe the required services and objectives within a project, associated components of digital data (models, model components and model properties) as well as interfaces and exchange requirements, ideally including information for the awarding/tendering of projects, thus avoiding unnecessary exchange iterations at data drops as well as interface problems between project participants. Further, it allows realistic adjustments of costs and prices.

According to (DIN Deutsches Institut für Normung e.V., 2019), the client sets relevant information standards within the framework of the Organization Information Requriements (OIR). These standards define how the exchange of information between individual stakeholders within a project should take place. This includes the exchange between the client and external companies/maintainers/stakeholders and the various contractors among themselves. Further, these defined standards specify how the information is to be classified or structured, and a method for assigning the depth of required information. Asset Information Requirements (AIR) set out the business, commercial and technical aspects of creating information in a project. The technical aspects here are understood to be the specification of the detailed information needed to meet requirements for building-related organizational information. AIR are significantly influenced in their content by the objectives of the overarching OIR. The third category, Project Information Requirements (PIR), is associated with project phases and thus milestones/delivery dates at which the contractor's services are due. The required construction target is also influenced by the higher-level OIR. Project scope, intended use of information, number of decision points (milestones), and information about decisions to be made, among other things, must be known in order to establish the PIR. Information Exchange Requirements (IER), which are established based on the PIRs and the AIRs according to BS 1192-2 and ISO 19650-2, define the business, commercial, and technical aspects of project information production. The technical aspects of the IER contain all the necessary information required to meet the PIR and are defined wherever information exchange processes occur. A fully comprehensive formulation of the IER is thus essential for lossless data and information exchange. Separate contractors define the exchange requirements for their respective trade or discipline, including internal exchange requirements between the individual trades/disciplines, which must be passed on to the BIM manager, e.g., using the BEP (whose structure and content are based on the EIR).

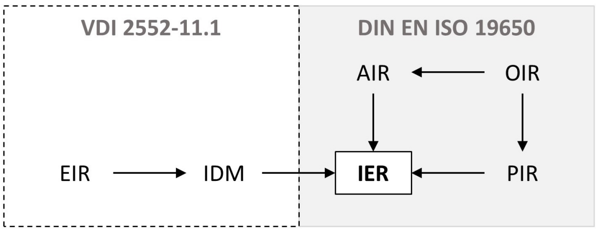

Figure 2: Role of the IER in information requirement specification process.

IER represent the key role in the BIM process for establishing object-oriented, geometric and semantic information in the form of requirements (VDI Verein Deutscher Ingenieure e.V., 2021b). From a data technology perspective, these requirements reflect all model components in the form of objects as well as their associated attributes, as depicted in Figure 2. In this context, attributes are object-related parameters that can assume variable values depending on the characteristics of the object. The presented description of IER is theoretic and information-centric. EIR and consequently IER are not the only approaches to describe information requirements. A brief description of similar concepts is given for placement in the following.

As part of the data-centric, process-oriented BIM process, the IER can be understood as part of the Information Delivery Manual (IDM) method, described in (DIN Deutsches Institut für Normung e.V., 2018), given the very specific and detailed structure of IER, which makes the individual objects and their attributes clearly identifiable. According to (VDI Verein Deutscher Ingenieure e.V., 2021a), the content of the IER specifies whether the information is actually within the scope of the selected process or is not relevant to it. This can generally be subdivided into three parts (header with administrative information, purpose, information requirements with information units). As part of the information requirements, the information units contain a description of the attribute to be mapped as well as information constraints related to a data type (data format), rules for using the concrete information unit, and a limited set of occurrences that the attribute can take (e.g., a list of values in enumerative form). These constraints may also contain special provisions on the use of the information. They describe the required rules for exchange and optionally a defined classification. The information units can be created software-independently with respect to their naming, content, form and other properties (See, R., Karlshoej, J., and Davis, D., 2012; VDI Verein Deutscher Ingenieure e.V., 2021a).

Similarly, the concept of Information Delivery Specifications (IDS) defines how objects, classifications, properties and values are delivered or exchanged with entities within the context of IDM (BuildingSMART International, 2020a,b). The first version of the XML-based IDS was published on the buildingSMART International GitHub repository in March 2021 (GitHub Community, 2021). Since January 2022, the second phase started the implementation of IDS in various software applications. Here, IDS should be able to be loaded into authoring tools in order to facilitate the creation, validation and concrete assignment of semantic information for both the users of the tools and the software tools themselves. The IFC-based open format (Industry Foundation Classes) can also accommodate other classification systems or domain extensions thanks to its flexible structure. The IDS standard aims to define computer-readable exchange requirements. This allows IFC files to be validated for builders and modellers in software applications, which performs automatic analyses. IER and IDS describe identical information in terms of objects and their attributes. While the IER focuses on the non-technical (non-computer readable) description, the IDS describes exact specifications of the objects in XML format usable for the IFC schema. Unlike IER, IDS are computer-readable and are processed automatically without human interference. Thus, the aforementioned automatic validation of IDS can be performed against defined IDM requirements. They provide a solution to develop predictable and robust data exchange workflows. IDS includes an open documentation and implementation interface, which closes with the flexible structure of the format to an extensible standard. In contrast to idsXML stands mvdXML, which is a format for coding Model View Definitions (MVD) and its sub-schemas. The main goal of mvdXML is to limit the entire IFC specifications to explicitly necessary parts for the respective MVD for software vendors. Given its' purpose, the mvdXML schema is very restricted and is designed to provide constraints on IFC specifications, while IDS allow more flexibility in their use. IDS have the great potential to represent all tasks of an IDM and the IER directly in machine-readable language, to perform automatic validation and verification processes and, in the context of MVD, to ensure a high quality standard of targeted data exchanges. They thus represent the future of embedding IDM and IER directly in a software environment.

A further development to define information requirements consists of defining the Level of Information Need (LOIN), defined in (DIN Deutsches Institut für Normung e.V., 2021). The goal of this concept is the definition and transfer of information requirements in an XML-based form. It contrasts to the IDS approach mainly in the applicability of the definition, since it is agnostic regarding the classification. Given their ongoing development, concepts such as Information Delivery Specification (IDS) and Level of Information Need (LOIN) were not regarded in this study.

Used Tools

The IER were defined using BIMQ, a web-based software that can be understood as a central dynamic attribute management system for models, from which information requirements and checking rules can be derived (AEC3 GmbH, 2023). Given its' interfaces, a direct transfer of the established requirements and attributes is possible for selected authoring tools (including Revit). An overview of the defined requirements is shown in Figure 3.

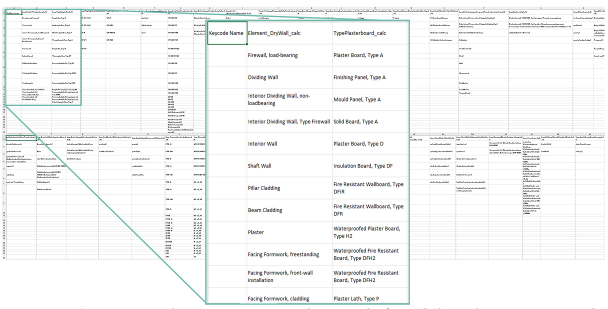

Figure 3: Exemplary excerpt of IER defined for this case study (dry wall elements and properties necessary for quantity take-off (QTO) generation, assigned to a key code).

The established authoring tool Autodesk Revit (version 2022) was used since it allows modelling, drawing, and listing for structures within a BIM process. Given their significance for the developed set of workflows for automated property supplement, the different types of parameters available in Revit are shortly explained as follows.

Classically, a Revit project consists of objects that are incorporated into the project via internal or loaded families, which in turn consist of parameters (or parameter sets) that can take on different characteristics depending on the family and the objects within it. They are divided into project parameters (for a specific project) and family parameters, which control the behaviour of the families/objects with variable values such as dimensions or materials and are only valid for the respective object/family as well as nested families. Further, family parameters are defined by a unique name, a GUID (globally unique identifier), a data type and a parameter group. These can be used to create different types of an object with the same parameters but different parameter characteristics (values of the parameters), thus allowing more flexibility. A further category is called shared parameters, which can be used in multiple projects or families. They are created and stored in a separate, independent file outside the project or family and are therefore protected against changes by users of the modelling environment. After assigning shared parameters to a project or family, they can be used as project or family parameters. Their use within a project is generally recommended because they are reusable, protected against changes, and have similar functionalities as the other types of parameters. They are defined by a unique name, a discipline, a data type, and can be organised into groups. Another useful tool are part lists, consisting of a tabular representation of information extracted from the inputs (columns) of the elements in a project. Each element is displayed in a row with the different values of the respective parameters from the columns and properties are automatically adjusted when changes are made in the model.

Part lists are always created for a category of objects (e.g., walls, ceilings, doors, etc.). A special form of part lists are key lists, where a key parameter is always automatically generated (and named). While part lists extract and compile data from a model, in key lists, data is defined by a user externally and entered into the model. They are only selectable, if their category corresponds to the category of the key table and can later be used as shared or project parameters. The parameters can then be filled with the values in the sense of a value list. If a key is selected, all parameters of the element that occurs in the key table are filled with values and locked before further processing. Key lists are therefore used for the uniform definition of component information, which is automatically added to the respective objects (Autodesk Inc., 2018). Thus, key lists are especially suitable for the automated import and assignment of IER data/properties into the modelling environment.

To address this automation, the visual programming API, Dynamo was used (Autodesk Inc., 2019). It is worth mentioning, that CPython3 was used as the Python version in addition to JavaScript and C. The packages (names; versions in brackets) Archi-lab.net (2022.212.322), bimorphNodes (4.2.1), BumbleBee (2021.25.3), Clockwork for Dynamo 2.x (2.4.0), Crumple (2022.2.13), Data-Shapes (2022.2.103), Orchid (212.1.0.8138), Rhythm (2022.4.3), Spring nodes (210.1.1) and Zebra (2016.7.2) were also used.

Implementation

The proposed workflow consists of a set of scripts that together allow the automated supplement of the defined IER data to the modelling environment using key lists, thus enabling/facilitating complete model-based bill of quantities (BOQ) generation in later applications. The description of this implementation is divided into data input and functionality as well as validation and results.

In a first step, the setup of an IER for the specific BIM use case 23 cost determination (5D planning) in connection with BIM use case 24 QTO must be defined. For this case

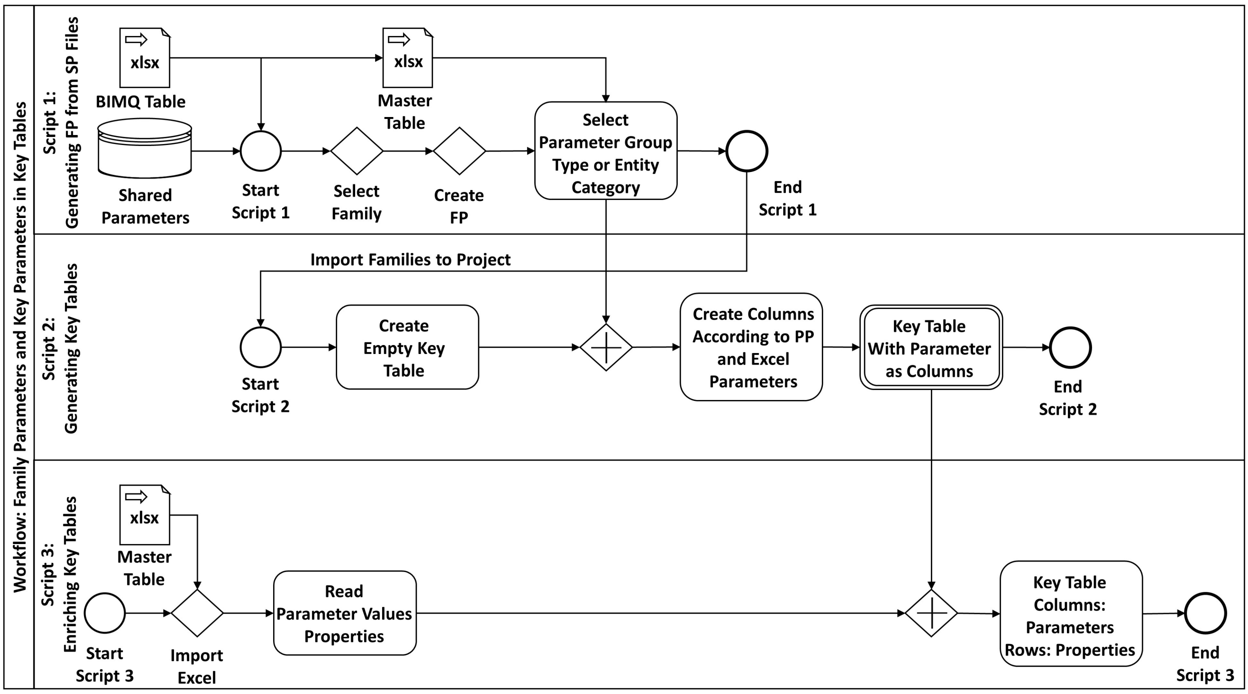

Figure 4: Implemented scripts.

study, this definition was done using the BIMQ application and then exported (using StLB-Bau classification and Autodesk Revit output), as previously mentioned, and depicted in Figure 3. This IER embodies the input data for the implementation.

Data Input and Functionality

The workflow, depicted in Figure 4, is composed of three parts. At the beginning of the workflow, the spreadsheet data is first processed using a master table to arrange the required information (property specifications for family parameters) in a predefined order, so that parameters and their characteristics can be generated in Revit from the exported BIMQ data (spreadsheet), which is imported into Revit and from which a shared parameter file is created. Project parameters are then derived from this shared parameter file, which in turn are used to build key tables. In a subsequent step, the key tables are enriched with the characteristics of the individual parameters. This provides the Revit user with a clear and structured list of parameters in the property records of objects, which can be filled with the stored value lists, e.g., via a drop-down menu. Furthermore, users can define their own keys in the key table, which assume selected values for their specific selection of parameters.

To execute the first script, the input spreadsheet must be preprocessed. For this purpose, a master spreadsheet (shown exemplarily in Figure 3; referred to as master table in Figure 4) was created containing three tables: MasterDynamo contains a list of all parameter names to be created in Revit in column 1. In column 2 each parameter can be assigned to a group defined in the shared parameter file. Column 3 contains the discipline of the individual parameter in a drop-down menu. In column 4, a parameter type can be selected from another drop-down menu for each parameter. Options for drop-down menus are stored in the MasterData table. In column 5, each parameter is assigned to a parameter group for project parameters. The latter determines under which property group the parameter appears on an object. Column 6 allows setting the parameter as a type parameter or a copy parameter (boolean operation). In the MasterKeyInput table, initially only one field is predefined in column 1 row 1 with the key name. This name is identical to the parameter key name in Revit for key tables. The appropriate list of values is then entered under each parameter. MasterKeyInput thus represents an identical match with the key table to be created in Revit.

With these three tables, two of which are to be filled and one of which is used as a database for selection options, the preparation of the input data is complete and the import into Revit can begin.

Three scripts have been developed in Dynamo for reading in the data from the table and creating family parameters with the associated value specifications based on assigned keys, which are explained as follows. Script 1 generates family parameters from shared parameter files. For this, all parameters from the master table are attributed to each selected family. When this script is executed, the spreadsheet that has just been processed is selected and loaded into the project (parameter names and information). All contents are read as strings and are available in individual lists (containing the spreadsheet data line by line). Afterwards, a comparison of existing shared parameters is necessary. All stored shared parameters (parameter names and properties) are loaded into another node, where they are compared to the names from the spreadsheet, resulting in an output list. Together with the input list of all parameter definitions of the shared parameters, a new list is defined, which contains all parameter definitions of the previously filtered parameter names. In a further step, files with the Revit family extension (rfa) are opened and searched using the DeepSearch functionality, capturing not only a folder with all Revit families, but also all sub-folders and their file paths.

Finally, all families under these file paths are opened consecutively and the (spreadsheet-based) shared parameters are added to the selected family parameters. In addition to the list of open families and parameter definitions, a list of associated parameter groups (formed from strings into objects of the parameter group of a list) and a list indicating whether the parameter is a copy or not (isInstance) are needed. This script is concluded with a loop to attribute all families with the family parameters and save them. The result of this script is an automatic attribution of all families with specific, predefined family parameters from generated shared parameter files, which in turn get their data from the input IER spreadsheet. In a subsequent process (Script 2), the supplemented families must be added to value lists in the form of key tables when they are loaded into a project. For this, an additional table with the value lists of the parameters in the form of the MasterKeyInput table is required (stored either directly in the processed table for the family parameters or elsewhere). The second script must reference this table and subsequently create an empty key table with all family parameters, which can then be filled with data from the previously mentioned value lists (parameters from the family parameters) in the third and final script, which reads the parameter values from the input data and assigns them to the generated key tables and thus to the Revit object properties.

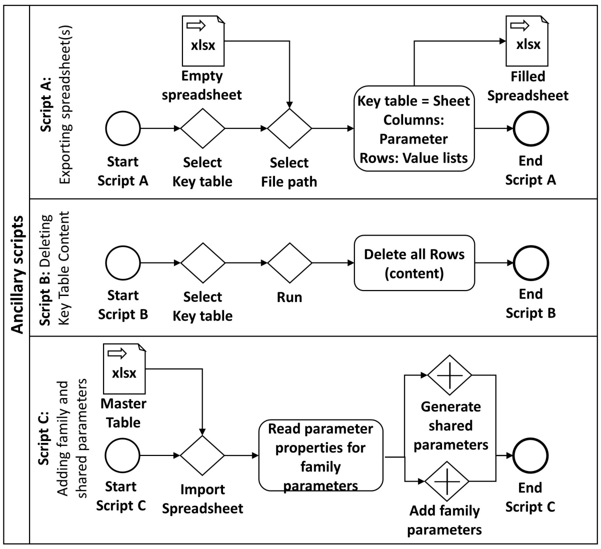

Figure 5: Ancillary scripts (A-C) to export key tables as spreadsheets, delete key table content and add additional shared parameters and family parameters.

Variations and ancillary scripts have been implemented in addition to the presented workflow, as depicted in Figure 5, thus offering additional functionalities, like exporting key tables from the modelling environment (script A), deleting content from the key table, thus enabling editing (script B) and adding shared parameters as well as family parameters from the master table (script C).

Validation and Results

To verify the results of the proposed automation, the model used for the case study and supplemented with requirements using the proposed workflow, was imported to a common TAI application iTWO (RIB Software GmbH, 2020). Therefore, the Revit model was exported using the iTWO Plugin, which converts the model into the native iTWO data format (cpiXML). The cpiXML model is then used for QTO and assignment of service descriptions, which can be exported and used for tendering and commissioning. The QTO based on the supplemented requirements done in the TAI application was compared to a QTO of the same model and the same requirements done directly in the Revit application. Thereby, it was verified, that the assigned requirements are correctly reflected in the exported data, thus confirming the functionality of the scripts and the objective of the proposed workflow.

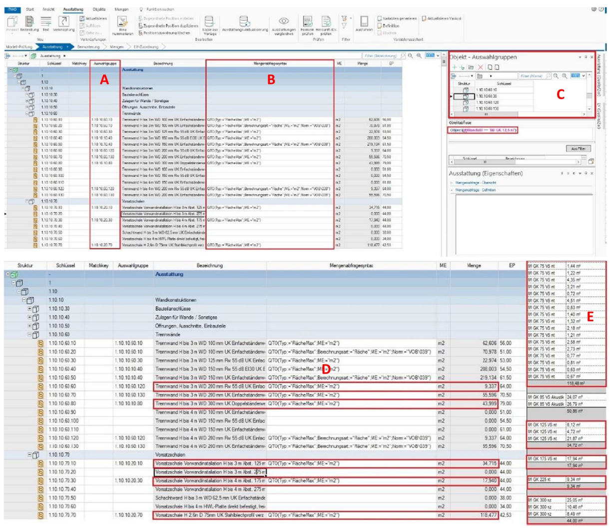

Figure 6: Validation screenshots (summarised). A, selected group (assigned keys); B, quantity query syntax (QTO); C, objects from selection group; D, Features + QTO + units; E, Revit QTO (reference for comparison).

The proposed automated supplement of standard TAI requirements within the modelling environment results in a simplified QTO procedure and description assignment, according to StLB-Bau classification, in the subsequent phase.

An overview of the available data (A-D) in the TAI application as well as the reference QTO used for comparison (E), is shown in Figure 6. It must be noted, that tendering requires more than a QTO, as described previously in Background, but for the verification of the set of scripts, this comparison of different QTOs is sufficient.

Given that tendering and commissioning of services is a crucial phase for successful construction projects, it is vital, that this phase can be handled in a model-based and automated manner. This possibility was illustrated by the proposed workflow.

Conclusion and Outlook

This study proposed a workflow consisting of defined input data and a set of scripts for automated supplement of TAI data to the modelling environment. With this approach, the continuous usage of the model in subsequent phases, such as tendering and commissioning, is facilitated. Furthermore, it results in a simplified, less error-prone and secured procedure of TAI document generation. The proposed workflow was validated in a case study, confirming the initial assumption.

The case study and the proposed workflow showed the added value of supplementing requirements to the modelling environment, by successfully adding TAI data for a specific classification to specific objects of an intended class in the model. The idea behind this study aims at opening up the discussion for a higher grade of input for modelling applications, such as Revit or ArchiCAD, by integrating functions to import and assign requirements as well as relevant information prior to or during the design/modelling phase. This study showed that this approach would be feasible and suggested a solution (within current possibilities of the chosen applications). Spreadsheets were used as input data, given the limitation of the Dynamo application. These were sufficient, given that spreadsheets offer semi-structured data. Nonetheless, other formats, such as XML, which is established for such requirement specifications (idmXML, idsXML and loinXML), offer more powerful tools, such as referencing and modularity of the requirement specifications themselves. Defining requirements using such tools and assigning them to a model (or objects within a model) would open up various possibilities for integrated design and offer interfaces to the Linked Data scope.

In future, the requirement supplement suggested in this study could be addressed either on a standard level by defining recurring requirements, or by supplementing project specific requirements to the modelling environment, depending on the stakeholder's goal or approach. Further future studies should address the favoured type of input for such integrated requirement tools or functions by addressing the question, if any of the differing standards currently under development (such as IDS and LOIN) or already established (IDM) are suitable for such an a priori supplement within the modelling environment.

References

Abbaspour, A., editor (2021). Digitales Bauen mit BIM: Use Case Management im Hochbau. bSD Verlag and Beuth Verlag GmbH, Berlin and Wien and Zürich, 1. auflage edition.

AEC3 GmbH (2023). BIMQ Documentation. https://bimq.centraldesk.com/en/categories/BkeX-documentation. Accessed: 2023-01-18.

ARGE BIM4INFRA2020 (2019). Handreichungen und Leitfäden - Teil 2: Leitfaden und Muster für Auftraggeber-Informationsanforderungen (AIA). https://bim4infra.de/wp-content/uploads/2019/07/BIM4INFRA2020_AP4_Teil2.pdf. Accessed: 2023-01-18.

Autodesk Inc. (2018). About Parameters | Revit 2018 | Autodesk Knowledge Network. https://knowledge.autodesk.com/support/revit/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Revit-Model/files/GUID-AEBA08ED-BDF1-4E59-825A-BF9E4A871CF5-htm.html. Accessed: 2023-01-18.

Autodesk Inc. (2019). Dynamo v1.3 Primer. https://primer.dynamobim.org/en/. Accessed: 2023-01-18.

Borrmann, A., Forster, C., Liebich, T., König, M., and Tulke, J. (2021). Germany's governmental bim initiative – the bim4infra2020 project implementing the bim roadmap. In Toledo Santos, E. and Scheer, S., editors, Proceedings of the 18th International Conference on Computing in Civil and Building Engineering, volume 98 of Lecture Notes in Civil Engineering, pages 452–465. Springer International Publishing, Cham.

British Standards Institution (2013). Specification for Information Management for the Capital/Delivery Phase of Construction Projects Using Building Information Modelling: [Incorporating Corrigendum no. 1; PAS 1192-2:2013.

BuildingSMART International (2020a). Machine readable Information Delivery Specification. https://technical.buildingsmart.org/projects/information-delivery-specification-ids/. Accessed: 2023-01-18.

BuildingSMART International (2020b). Technical Roadmap buildingSMART: Getting ready for the future.

Deubel, M., Wolber, J., and Hagsheno, S. (2018). Identifikation, Analyse und kategorisierung von bim-anwendungsfällen/identification, analysis and categorization of bim use cases. Bauingenieur, 93(07-08):295–303.

DIN Deutsches Institut für Normung e.V. (2018). Bauwerksinformationsmodelle - Handbuch der Informationslieferungen Teil 1: Methodik und Format: Teil 1: Methodik und Format; DIN EN ISO 29481-1:2018.

DIN Deutsches Institut für Normung e.V. (2019). Organisation und Digitalisierung von Informationen zu Bauwerken und Ingenieurleistungen, einschließlich Bauwerksinformationsmodellierung (BIM) Informationsmanagement mit BIM: Teil 2: Planungs-, Bau- und Inbetriebnahmephase; DIN EN ISO 19650-2:2019.

DIN Deutsches Institut für Normung e.V. (2021). Bauwerksinformationsmodellierung – Informationsbedarfstiefe – Teil 1: Konzepte und Grundsätze; Deutsche Fassung EN 17412-1:2020.

Eastman, C., Teicholz, P., Sacks, R., and Lee, G. (2018). BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers. John Wiley & Sons, Hoboken, New Jersey, 3. ed. edition.

GAEB - Gemeinsamer Ausschuss Elektronik im Bauwesen (2020). About Us - Main Focus. https://www.gaeb.de/en/about-us/#1557004565364-68d934f3-b0f2. Accessed: 2023-01-18.

GitHub Community (2021). buildingSMART/IDS: Computer interpretable (XML) standard to define Information Delivery Specifications for BIM (mainly used for IFC). https://github.com/buildingSMART/IDS. Accessed: 2023-01-18.

Mellenthin Filardo, M. (2019). Praxisrelevante Auftraggeber-Informations-Anforderungen für Infrastrukturprojekte als Indikatoren des Einführungsgrades von BIM in Deutschland. In Sternal, Maximilian; Ungureanu, L.-C. B. L. B.-G. C., editor, Proceedings des 31. Forum Bauinformatik, pages 9–16. Universitätsverlag der TU Berlin.

Mellenthin Filardo, M. and Krischler, J. (2020). Basiswissen zu Auftraggeber-Informationsanforderungen (AIA). BIM Basics. bSD Verlag, Berlin.

RIB Software GmbH (2020). RIB iTWO: Version 2020. https://www.rib-software.com/en/news-events/software-news/details?tx_news_pi1%5Bnews%5D=854&cHash=ff3779d032500114a905ea681ff319f0. Accessed: 2023-01-18.

Roquette, A. J. and Schweiger, D., editors (2020). Vertragsbuch Privates Baurecht: Kommentierte Vertragsmuster. Beck-online Bücher. C.H. Beck, München, 3., vollständig überarbeitete auflage edition.

Rösel, W. and Busch, A. (2008). AVA-Handbuch: Ausschreibung - Vergabe - Abrechnung. Vieweg + Teubner, Wiesbaden, 6., überarb. aufl. edition.

Schwerdtner, P. (2018). Nutzung von BIM in der Angebotsbearbeitung. Bautechnik, 95(3):222–228.

See, R., Karlshoej, J., and Davis, D. (2012). An Integrated Process for Delivering IFC Based Data Exchange. https://standards.buildingsmart.org/documents/IDM/IDM_guide-IntegratedProcess-2012_09.pdf. Accessed: 2023-01-18.

VDI Verein Deutscher Ingenieure e.V. (2021a). Richtlinie VDI 2552 Blatt 10: Building Information Modeling - Auftraggeber-Informations-Anforderungen (AIA) und BIM-Abwicklungspläne (BAP).

VDI Verein Deutscher Ingenieure e.V. (2021b). Richtlinie VDI 2552 Blatt 11.1: Building Information Modeling - Information Exchange Requirements for BIM Use Cases.